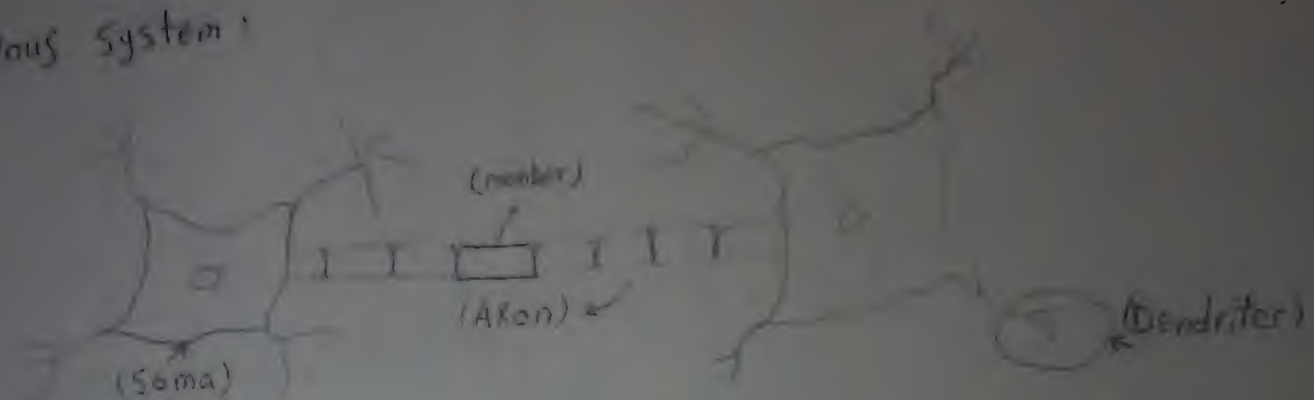
Chapter (I) : From biological to ANN (Artificial Neural Network)

* Nervous system :



- Soma : (generate Tted signals)
- Axon : (Carrying signals (ions))
* chemical reactions (ion - Pumb) occurs in Axon

Ka - - - - - + + + + - - - - -

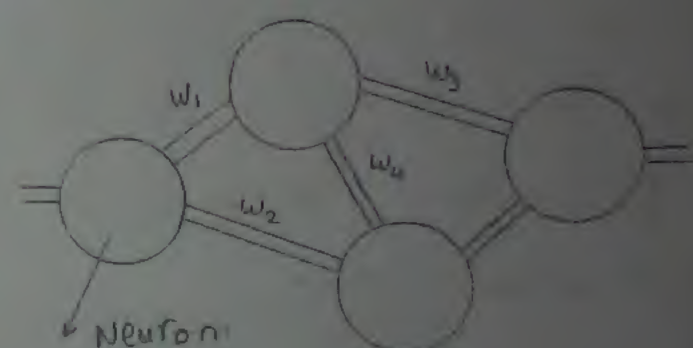
(ionisation) inside the Axon's members

K

+ + + + + + - - - - - + + + + +

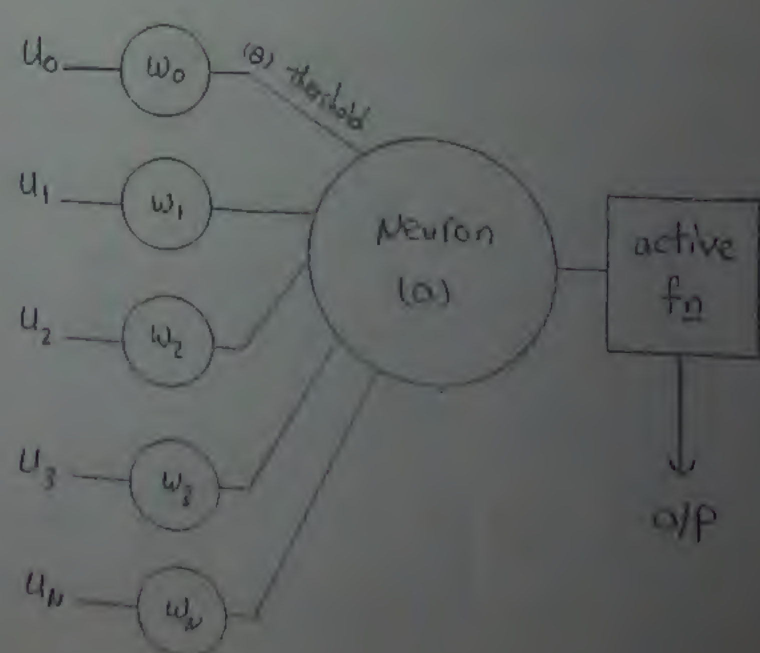
* Neural Network :

- Neuron ⟶ Soma
- Layers ⟶ Axon
- Thershold } ⟶ chemical reaction
- weight



ANN : (Art. Neural Network)

- I/P layer
- Hidden layer (Neurons)
- O/P layer



* Active function :

$$a = \sum_{n=1}^{n=N} u_n w_n + \text{Thershold}$$

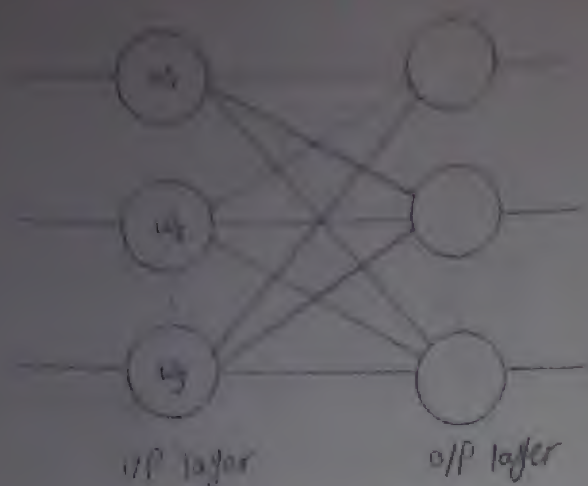
$$= \sum_{n=0}^{n=N} u_n w_n \quad \text{(with thershold inside)}$$

$$\theta = u_0 w_0$$

* (in matrix)

$$a = u^T w + \theta$$

Architecture of NN:



signal layer Perceptor

multi layer Perceptor

* ω is adapted according thershold (activation function)

Learning:

- Supervised Learning (training for NN)
- unsupervised Learning (No i/P - o/P relation)
- Reinforcement learning (No i/P : the i/P generated inside NN)

Activation Function:

- A relation between o/P and (i/Ps & weights)

* Cost function : Compare between measured output & actual output
  - min. cost fn ⟶ optimal Solution.

$C = E[f(x) - y]$ (The mean value of the difference between measured o/P & actual o/P)

* NN may be trapped at C_5 : think it the best sol. but if it continue to C_8 is the best sol. (NN Problem)

* to get the optimal sol. we take C gradient to ω $\frac{\partial C}{\partial w} \cong 0$

* For Ideal Solution, (Dynamic states) (phase space)

- Convergance

o/P Convergace to ideal state

- cyclic (oscillating)

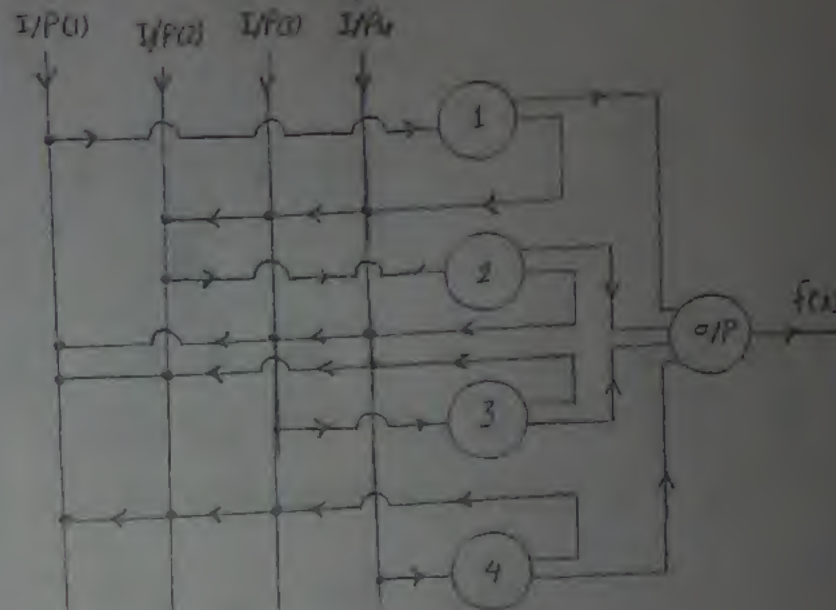
oscillate and return to same point

- chaotic

get far from ideal solution

Recurrent NN : (Chapter II)

- o/P of each N backwards to all components expect itself.
- $w_{ii} = 0$
- $w_{ij} = w_{ji}$ (symmetry)



Applications of NN :

- Classifications (Pattern Recognition) (Image signal processing)
- Data processing
- Control
- Robotics
- Medical applications
- speech Recognition
- Picture Recognition

note: the main Parameter is the weight of each interconnection, based on application & activition function.

- output is function in activition function → $X(t) = f(a)$
- activition fn: Linear, Thershold (or), ramp

the target from RNN

- unique solution
- Existing solution
- stable state

$\left.\begin{matrix} \\ \\ \end{matrix}\right\} x(t) \longrightarrow \frac{d\,x(t)}{dt} = F(X(t))$

$= F(X_1(t), X_2(t), \ldots, X_n(t))$

* unique solution $\longrightarrow$ restriction
(certain condition to reach st. state for each Neuron not for whole Network (w adaptive))

$x_1(t_1) \longrightarrow w_1$
$x_1(t_2) \longrightarrow w_2$
$\vdots$
$x_1(t_n) \longrightarrow w_n$

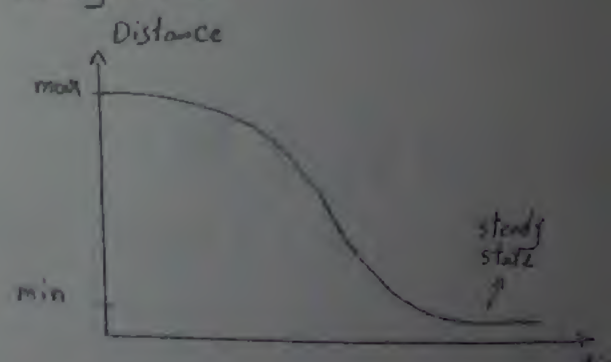
$\Longrightarrow$ distance $= |x_1(t_2) - x_1(t_1)|$
$= |X_1(t_3) - X_1(t_2)|$

* operating Network in different times (iteration)

* if distance decrease, we approaching unique solution

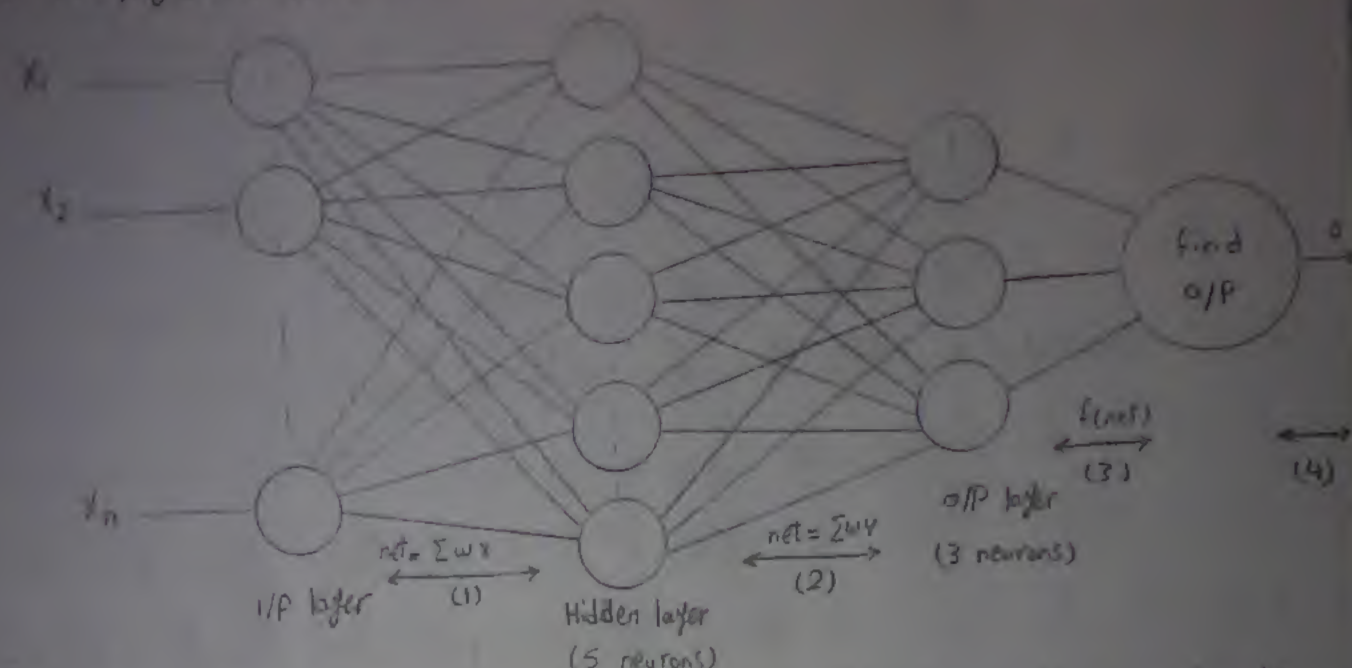
$d \longrightarrow \underset{(0)}{\min} \longrightarrow$ toward ideal sol. $\longrightarrow$ steady state

- After we reach stedy state, we start testing & Validation.
- If we get S.S far from zero, so if we change time weights will not change
- Return and reconstruct Network.



note: Large no. of neurons cause more Processing, more time to get S.S.
(NN Problem)

Back Propagation Neural Network (BPNN)



- The relation between i/p ⟹ layers ⟹ o/p comes from the activation function:

* i/p x_d , o/p t_d for neuron j , it will receive the following function:

$$net_j^d = \sum_{k=1}^{5} w_{jk}\, x_k^d \quad \text{— (1)}$$

$$V_j^d = f(net_j^d) = f\left(\sum_{k=1}^{5} w_{jk}\, x_k^d\right) \quad \text{— (2)}$$

* o/p (at neuron i)

$$net_i^d = \sum_{j=1}^{3} w_{ij}\, V_j^d \quad \text{— (3)}$$

$$= \sum_{j=1}^{3} w_{ij} \cdot f\left(\sum_{k=1}^{5} w_{jk}\, x_k^d\right) \quad \text{— (3)}$$

* final o/p (O): $O_i^d = f(net_i^d) = f\left(\sum_{j=1}^{3} w_{ij} \cdot f\left(\sum_{k=1}^{5} w_{jk}\, \underbrace{x_k^d}_{i/p}\right)\right)$ — (4)

note:

* In Traditional NNs, the o/p changes to reach ideal case only by try & Error.
- Nowadays, there are another techniques like optimization, genetic algorithm, ...) To obtain the ideal (optimal) weights for this network to reach or be close to ideal solution.

$$E[w] = \frac{1}{2} \sum_{d=1}^{N} \sum_{i=1}^{J} (t_i^d - O_i^d)^2$$

desired o/p — actual o/p

then $\frac{\partial E}{\partial w}$ to determine the weights of minimum energy

$E = f(o - T)$ → our target is minimum energy

- Back-propagation Algorithm for m # of layers:

(1) initialise the weights to small random values.

(2) choose pattern (X_K^d) and apply it to the i/p layer $V_K^0 = X_K^d$ for all K

(3) propagate the signal through the Network ⇒ $V_i^m = f(\sum_j^n w_{ij}^m V_j^{m-1}) = f(net_i^m)$

Preceding layer o/p

(4) Compute delta for the output layer

$$\delta_i^m = \dot{f}(net_i^M)(t_i^d - V_i^M)$$ ⟹ related to cost f_n (Energy $\underline{fn}$)

M: # of layers; error; derivative; target; actual

(5) Compute δ at Precending layers

$$\delta_i^{m-1} = \dot{f}(net_i^{M-1})(\sum w_{ij}^m \delta j^m)$$

(6) update the Connections (weights)

$$\Delta w_{ij}^m = \eta \delta_i^m V_j^{m-1}$$ ⟹ δw_i → learning rate

$(w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij})$ ⟹ new weight

Radial Base function Network (RBFN)

- Special case of BPNN when activation function is radial Basis

* Advantages of RBFN :

- No Local minimum
- No oscillation
- trained rapidly

* Disadvantages of RBFN :

- After training it becomes slow.

Basis function : $\phi = \underbrace{|x_i - x_j|}_{\text{distance (euclidien distance)}}$

. i/p it self.

$\phi(x) = \sum_{i=1}^{N \to N\ \text{space}} w_i \; (|X_i - \underbrace{C_i}_{\text{centre}}|)$

The activation function in radial Basis.

Types of Radial :

- Gaussian
- Multi-Quadric
- Inverse-Quadric
- Inverse multi-Quadric

must be symmetric (distance between Point and other Point must be the same)

Pattern Recognition (Face Recognition using NN)

* Face Recognition: A science based on image Processing

  - image ⟶ image Processing ⟶ features

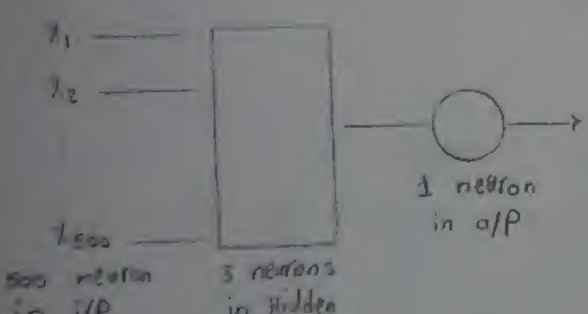
features
- Color
- shape
- age
- length
- ⋮

* main Procedures:
- Capture the image
- Image enhancement
- Features Extraction
- Features selection (significant feature

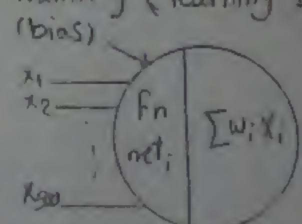
Example: Picture with 500 pixels.
- the main feature is color intensity of pixel

0 Black ⟷ 255 white

- 500 i/P neurons (no. of pixels), 1 o/P neurons (recognition u, d, R, L)



(1) Training (learning by an algorithm (BPNN))



(2) Validation (Test from data base)

(3) Test (test out of data base)

* up ⟶ down (false)
  up ⟶ up (true)
  up ⟶ Right (False

⟶ For some situations (no. of false = no. of True) we can calculate Confusion matrix & Then The accurcy.

Examples:
- Classification between a Car and a Van
- Classification between happy, sad and natural face
- Classification ... direction of a face

- What is Neural Network mean? (main Idea / Concept)

Neural Networks (NNs) also known as Artifical Neural Networks (ANN), are massivaly Parallel interconnectio network of simple (usually adaptive) elements and Their hierarchical organization which are intended to interact with objects of real world in the same way as biolgical nervous system do.